

## Procesarea grid

Prep. Felician ALECU

Catedra de Informatica Economica, A.S.E. Bucuresti

*Grid computing represents unlimited opportunities in terms of business and technical aspects. A grid is a collection of machines, sometimes called as nodes, resources, members, donors, clients, hosts, engines and so on. The goal is to create the illusion of a powerful computer out of a large collection of connected systems sharing resources. Some resources may be used by all users of the grid while others may have specific restrictions. The most common resource is computing cycles provided by the processors. There are three primary ways to exploit the grid network. The first is to run an existing application on an available machine on the grid rather than locally. The second is to use an application designed to split its work in such a way that separate parts can be executed in parallel on different processors. The third is to run an application that needs to be executed many times on many different machines in the grid.*

**Keywords:** *Grid computing, grid network, parallel processing, scalability, queuing theory, execution queue, scheduling.*

**P**rocesarea de tip grid reprezinta un nou nivel de evolutie a procesarii distribuite. Procesarea grid creeaza utilizatorilor impresia existentei unui singur calculator virtual puternic în conditiile în care sistemul este de fapt format prin interconectarea mai multor sisteme de calcul eterogene care partajeaza o serie de resurse. Un astfel de comportament poarta numele de uniprocessor virtual datorita faptului ca utilizatorii nu trebuie sa se preocupe de aspecte legate de localizarea resurselor, protocoalele de comunicatie folosite, modalitatile de planificare utilizate etc.

Standardizarea mecanismelor de comunicatie dintre sisteme eterogene a condus la explozia internetului. Interconectarea unor sisteme individuale si standardizarea modalitatilor de folosire a resurselor partajate vor conduce cu siguranta la explozia procesarii de tip grid.

Acest tip de procesare reprezinta o modalitate de îmbunatatire a modului de utilizare a resurselor de calcul prezente în retea. În marea majoritate a organizatiilor exista resurse de calcul slab utilizate (procesor, memorie, discuri etc.). Calculatoarele de tip desktop, de exemplu, au o rata de utilizare mai mica de 5%. Eficienta utilizarii unor astfel de resurse poate creste simtitor prin folosirea procesarii de tip grid.

Cel mai simplu mod de exploatare a unei retele grid este executia unei aplicatii pe un alt

calculator decât acela pe care este rulata în mod curent. Daca calculatorul pe care este rulata în mod traditional aplicatia este ocupat cu executia alte taskuri, acesta poate declansa de la distanta executia aplicatiei respective pe un alt calculator disponibil din retea si care îndeplineste toate cerintele hardware si software cerute de program.

O categorie de aplicatii care se preteaza foarte bine pentru procesarea grid este formata de programele care prelucreaza volume mari de date pentru obtinerea rezultatelor dorite. Aplicatiile pot fi rulate în conditii de eficienta în cadrul retelei grid prin realizarea unei partitionari în spatiul datelor. Fiecare calculator din retea grid va opera doar asupra unui subset al datelor de intrare. Rezultatul final al prelucrării se va obtine prin combinarea rezultatelor de la nivelul elementelor de procesare. Subseturile de date pot fi replicate în retea, utilizând capacitatea de stocare disponibila si neutilizata la nivelul retelei grid. Partitionarea se poate face si în spatiul instructiunilor. Programul va fi astfel divizat în taskuri independente care pot fi rulate în paralel pe procesoarele retelei.

O a treia varianta de exploatare a procesarii de tip grid o reprezinta executia în paralel, pe procesoarele disponibile, a unei aplicatii care trebuie sa se execute de mai multe ori. Procesarea de tip grid poate conduce la o mai buna

echilibrare a modului de utilizare a resurselor deoarece aplicatiile pot fi rulate pe un alt calculator liber din retea în cazul în care cel pe care trebuiau executate nu este liber.

Rețele de tip grid ofera un suport deosebit pentru procesarea paralela prin utilizarea simultana a mai multor elemente de procesare. În felul acesta se obtine o putere de calcul sporita care este folosita de aplicatii pentru minimizarea timpului în care sunt obtinute rezultatele dorite. Procesarea de tip grid, datorita faptului ca foloseste resurse deja existente dar neutilizate, reprezinta o alternativa mult mai putin costisitoare a calculatoarelor paralele. Puterea de calcul obtinuta prin interconectarea mai multor sisteme de calcul a deschis noi orizonturi, nu numai în domeniul stiintific dar si în domenii cum ar fi biomedicina, modelarea financiara, explorarea petroliera, animatia video si multe altele.

Pentru ca un program sa poata fi rulat în cadrul unei rețele de tip grid este necesar ca acesta sa poata fi divizat în sarcini de calcul independente care sa fie executate în paralel. Cu cât taskurile care compun programul comunica mai puțin între ele cu atât aplicatia devine mai scalabila. Un program perfect scalabil ar fi de 10 ori mai rapid în cazul în care ar fi rulat pe 10 procesoare fata de cazul în care s-ar executa într-un sistem uniprocessor. Scalabilitatea este masura eficientei cu care sunt utilizate procesoarele din cadrul rețelei de tip grid.

Desigur ca exista o serie de bariere care stau în calea scalabilitatii totale cum ar fi numarul de taskuri în care poate fi divizat programul sau existenta unor taskuri care nu sunt independente unele fata de altele, fiind necesar ca ele sa comunice între ele si sa se sincronizeze. De asemenea, mecanismele de comunicare si sincronizare implementate la nivelul rețelei grid pot sta în calea atingerii scalabilitatii totale.

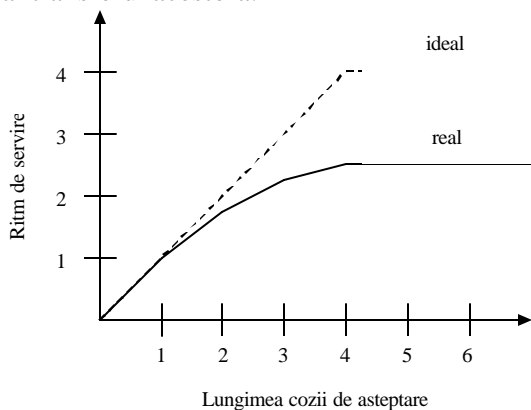
Daca privim rețeaua grid prin prisma teoriei asteptarii, cele  $n$  procesoare din sistem reprezinta de fapt posturile statiei de servire. Daca numarul de procese din sistem este mai mare ca  $n$ , atunci toate procesoarele vor fi ocupate iar ritmul de servire va atinge valoarea maxima. Daca însa în sistem vom avea un nu-

mar de procese mai mic ca numarul posturilor de servire, anumite procesoare vor fi inactive, ceea ce va conduce catre o scadere a ritmului real de servire. Se poate concluziona ca ritmul real de servire nu este constant însa acesta depinde de numarul de consumatori din sistem. În figura 1, pentru o rețea grid compusa din patru calculatoare sa exprimat ritmul real de servire în functie de lungimea cozii de asteptare. Din figura reiese faptul ca ritmul de servire creste odata cu lungimea cozii de asteptare pâna în momentul în care toate cele patru procesoare sunt ocupate. Din acest moment, orice crestere a dimensiunii firului de asteptare nu mai atrage dupa sine si o îmbunatatire a ritmului de servire. Diferentele care se pot remarca între ritmul real de servire si cel ideal sunt datorate barierelor discutate anterior.

Aplicatiile cu cerinte intensive de calcul care au fost proiectate pentru a fi executate pe calculatoare paralele vor fi foarte usor transformate pentru a putea rula în cadrul unei rețele de tip grid. Din pacate însa nu toate aplicatiile pot fi transformate pentru a fi în paralel executate în rețele grid. La ora actuala nu exista pe piata instrumente care sa transforme automat codul secvential în cod paralel executabil în rețele grid. Exista însa pachete care ne asista în crearea aplicatiilor paralele orientate grid.

Rețelele grid pun laolalta resursele calculatoarelor independente sub forma unei singure resurse virtuale. Din acest motiv ele pot contribui la echilibrarea modului de utilizare a resurselor de calcul. Daca se înregistreaza un vârf de activitate, planificatorul poate schimba politica de alocare a taskurilor pe procesoarele componente schimbând de fapt disciplina de servire în cadrul sistemului de asteptare format. Astfel, se pot planifica taskuri pe masinile care au un grad de utilizare mai scazut sau se pot asocia prioritati taskurilor deja existente pentru ca cele vitale sa fie executate înaintea celor mai puțin importante. Planificarea taskurilor poate fi facuta astfel încât taskurile sa comunice cât mai puțin si pe distante cât mai scurte. În plus trebuie sa fie implementate, la nivelul întregului sistem, mecanisme de comunicare si sincronizare între

taskuri pentru a preveni cazuri în care date partajate sunt modificate concomitent de mai multe procese afectând astfel integritatea acestora. Reteaua de interconectare este și ea un factor care poate afecta performanțele rețelei datorită faptului că nu întotdeauna datele de care are nevoie un task se găsesc în nodul pe care acesta a fost planificat fiind necesar transferul acestora.



**Fig. 1.** Ritmul de servire exprimat în funcție de dimensiunea cozii de așteptare

În momentul de față, în cadrul sistemelor critice elementele hardware sunt duplicate pentru a se asigura în acest fel protecția sistemului împotriva unei eventuale defectiuni care ar putea afecta buna funcționare. Pe viitor, o astfel de metodă hardware de protecție va fi înlocuită cu una pur software prin utilizarea rețelelor grid. În acest fel costurile ar putea scădea foarte mult. Planificatorul va fi responsabil de planificarea taskurilor doar pe nodurile valide din sistem. Pentru sistemele critice acesta ar putea lansa multiple copii ale unui task pe mașini diferite pentru a preveni situația în care unul dintre sisteme devine indisponibil.

Progresele rapide înregistrate în domeniul comunicațiilor fac procesarea de tip grid posibilă. De performanțele sistemului de comunicație depind rezultatele care pot fi obținute la nivelul întregului sistem. Astfel, unele taskuri necesită largi volume de date care de cele mai multe ori nu se află în nodul respectiv. Într-un astfel de caz lățimea de bandă devine o resursă critică care poate limita performanțele întregii rețele grid.

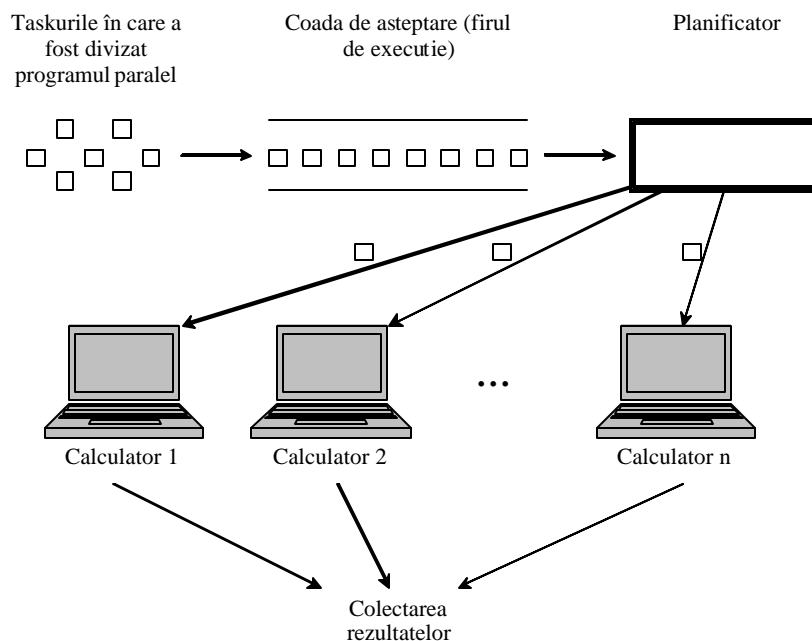
Procesarea de tip grid a devenit din ce în ce mai populară în ultimii ani datorită faptului că s-au obținut rezultate spectaculoase prin utilizarea unei puteri de calcul foarte mari care a rezultat din interconectarea unor calculatoare obișnuite în cadrul unei rețele grid. Acest caz aparține de exploatarea a paralelismului reprezintă un alt domeniu în care teoria așteptării își găsește o largă aplicabilitate. Programul paralel este divizat în sarcini de calcul care sunt distribuite unor calculatoare distincte din cadrul rețelei. Sistemul paralel este format de această dată din sisteme distincte și eventual eterogene legate între ele în rețea iar programele beneficiază din plin de multiplicitatea resurselor (procesoare, memorie etc.). Rezultatele generate de taskurile executate în paralel sunt colectate și combinate pentru obținerea rezultatului final. În figura 2 se prezintă modul în care este executat un program în cadrul unei rețele grid.

Potentialul deosebit în ceea ce privește procesarea paralelă este una dintre cele mai importante caracteristici ale sistemelor grid. Acest lucru a permis deschiderea de noi frontiere nu numai în domeniul științific, ci și în ceea ce privește modelarea financiară, exploatarea petrolului, desene animate și așa mai departe. În toate aceste cazuri sunt folosiți algoritmi care pot fi divizați în părți independente din punct de vedere al execuției. Task-urile sunt executate pe calculatoarele care formează rețeaua grid. Teoretic se poate presupune că prin utilizarea unui număr de 10 procesoare se va obține un timp de execuție de 10 ori mai mic. În practică se înregistrează valori mult mai mici, deoarece apar costuri de calcul suplimentare datorate comunicării și sincronizării dintre procesoare. Rețeaua grid este responsabilă pentru trimiterea unui task pentru a fi executat de către o mașină din rețea. Planificatorul este acela care va găsi cea mai apropiată mașină care să poată executa taskul respectiv. Atunci când un nod devine liber, acesta va raporta nodului responsabil de managementul rețelei faptul că așteaptă noi însărcinări.

Planificatoarele sunt în general sensibile la modificarea resurselor disponibile în cadrul rețelei grid deoarece astfel de schimbări pot

afecta timpul estimat de executie al aplicatii. De asemenea, planificatorul va trebui sa tina cont de faptul ca o serie de resurse pot fi re-

zervate în avans de catre anumite aplicatii din dorinta de a fi respectate termene limita si de a garanta calitatea serviciilor.



**Fig. 2** Executia unui program în cadrul rețelei grid

Planificatoarele implementează adesea sisteme de prioritati. Astfel, sunt menținute comitent mai multe fire de executie, câte unul pentru fiecare clasa de prioritati, iar atunci când o masina din rețea devine libera, acestea îi va fi alocat un task extras din coada de asteptare aferenta clasei cu prioritatea cea mai mare.

Desigur ca și procesarea grid are anumite limite. Nu toate aplicatiile se pretează a fi executate în cadrul unei rețele de tip grid. Unele programe pur și simplu nu pot fi paralelizate, iar pentru altele este nevoie un volum de munca impresionant pentru a fi rescrise sau modificate, iar de cele mai multe ori acest efort nu se justifică ținând cont de câștigul de viteză obținut.

### Bibliografie

- [Asm03] S. Asmussen, *Applied Probability and Queues*, Springer, 2003
- [Dod02] Gh. Dodescu, B. Oancea, M. Raceanu, *Procesare paralela*, Editura Economica, 2002
- [Tan99] A. S. Tanenbaum, *Organizarea Structurata a Calculatoarelor*, Computer Press Agora, 1999
- [Sab95] G. W. Sabot, *High Performance Computing*, Addison-Wesley, 1995